;login:

The USENIX Association Newsletter

Volume 7 Number 5 November 1982

CONTENTS

UNICOM — The Winter Technical Meeting

The next general meeting of the USENIX Association will be a joint meeting with /usr/group and the Software Tools Users Group. It will be called UNICOM and will be held in San Diego, California, Tuesday, January 25 through Friday, January 28, 1983, at the Town and Country Hotel. The meeting is being hosted by the University of California at San Diego, with Tom Uter acting as the overall local conference coordinator.

The local arrangements chairperson is

Judy DesHarnais
UNICOM
P.O. Box 385
Sunset Beach, CA 90742
(213) 592-3243

The meeting registration packet was mailed to an extensive list November 22. If you do not receive the registration packet or need registration and/or accommodation information contact Ms. DesHarnais.

The deadline for submissions for the December issue of *;login:* is December 17

The Toronto Meeting

The Summer, 1983, technical meeting of the USENIX Association will be held in Toronto, Ontario, Canada, the week of July 11. The host of the meeting will be Human Computing Resources Corporation.

Boston Meeting Proceedings

The Boston Meeting conference proceedings will be available in January. They are over 350 pages long and will cost about $25. The proceedings will be available at UNICOM and by mail from /usr/group; specific ordering information will be printed in *;login:* when it is available.

AT&T Announces System V

AT&T has announced that it plans to license **and support** System V, their latest in-house version of UNIX*, early next year.

System V will be offered as a machine-independent "standard" operating system for 16 and 32 bit computers. It will run on PDP-11 and VAX machines; AT&T has not announced what other CPU's it will run on, if any. It is reported to have improvements in the areas of performance, screen editing, text processing, file system maintenance, and communications.

Several levels of support will reportedly be offered including a hot-line service, consultation, technical seminars, newsletters, electronic mail to report problems, and periodic releases to correct problems.

Details will be announced in January, 1983, and discussed at UNICOM.

DEC Announces Supported UNIX V7

On December 6 at the DECUS meeting in Anaheim, California, Digital Equipment Corporation announced that they will be offering their modified UNIX Version 7 for PDP-11's as a standard DEC product. It will be orderable just as any other DEC product, with formal field service and software services (i.e., **support**) available. The product name is "V7M-11". The announcement stated that it will be offered as a binary sub-license only.

In the announcement DEC identified the features distinguishing their product from that offered by AT&T as being:

(1) support will be available;

(2) it runs on all memory-managed PDP-11 models and is fully customer installable;

(3) it supports all current PDP-11 peripherals;

(4) it includes code for bad disk block handling;

(5) it includes a complete error logger and user-mode diagnostics; and

(6) it includes the U.C. Berkeley *vi* editor.

V7M-11 will be available about May, 1983.

*UNIX is a trademark of Bell Laboratories.

Performance Issues of *VMUNIX* Revisited

Preface

At the request of the editor, I have forwarded the following report entitled "Performance Issues of *VMUNIX* Revisited". Regretably, this report is now somewhat out of date and also depends heavily on three other reports not included here:

1) An unpublished report by David Kashtan (circa 1979) entitled "UNIX and VMS - Some Performance Comparisons".

2) An unpublished report by William Joy (circa 1979) entitled "Comments on the Performance of UNIX on the VAX".

3) An unpublished internal memorandum from April 1981 criticizing both the virtual memory performance of VMUNIX and the performance of the VAX F77 compiler.

Unfortunately, without these other reports, much of what is contained below cannot be put into proper perspective. Nevertheless, it was felt that it would be of general benefit to include the following in *;login:*.

Thomas Ferrin
23 June 1982

Performance Issues of *VMUNIX* Revisited

Thomas E. Ferrin

Computer Graphics Laboratory
School of Pharmacy
University of California
San Francisco, California 94143

7 May 1981

The 20 April 1981 memorandum from *<deleted by request>* discussed several VAX performance issues and reached conclusions very unfavorable toward the *UNIX* Operating System. Unfortunately, many of the measurements quoted for *VMUNIX* were either inaccurate or out of date. Since such measurements must be the foundation for the recently proposed change in operating systems, it is important to have the most up-to-date results. The *VMUNIX* measurements cited below were performed during the week of 24 April to 1 May 1981 on a VAX 11/780 computer running version 4.1 of *VMUNIX* with 2.0 Mb of memory. The *VMS* measurements are the same as those quoted from the memorandum of 20 April 1981.

There are two separate performance issues. The first deals with the operating system "paging" algorithm, certainly a crucial area in a virtual memory environment. The second issue deals with Fortran performance. These are summarized as follows:

1. **"Paging" performance.** Recent enhancements to *VMUNIX* have improved its paging performance considerably. The addition of a new advisory call (*vadvise(SEQL)*) to the operating system to inform it that a program will be exhibiting strongly sequential behavior and thereby increasing the

program's "page-in cluster factor"* under such circumstances (in addition to other changes) have yielded the following improvements:

Table I
Sequential Page Access

	VMS	*VMUNIX 4.0*	*VMUNIX 4.1 w/o vadvise(SEQL)*	*VMUNIX 4.1 w/vadvise(SEQL)*
CPU Time	39.0	1:07.0	1:17.2	1:17.2
Real Time	4:32.0	19:17.0	4:56.0	3:17.0

Thus, a strongly sequential program runs faster (even out-performing *VMS* in real time) if the system is told that the program has sequential access behavior. Soon it may not even be necessary for a program to inform the system it is sequential, since such behavior can be detected automatically by the system. The significance of sequential memory behavior is referred to again below.

Of course sequential access is not the only type of memory reference behavior. The other benchmarks referred to in the 20 April memo are quoted below:

Table II
Random Page Access

	VMS	*4.0 VMUNIX*	*4.1 VMUNIX*
CPU Time	31.0	51.9	1:09.6
Real Time	6:00.0	12:43.0	6:51.0

Table III
Gausian Page Access

St. Dev.	*VMS*		*4.0 VMUNIX*		*4.1 VMUNIX*	
	CPU	*Real*	*CPU*	*Real*	*CPU*	*Real*
1	15.5	:16	23.0	:23	14.7	:15
10	17.8	:18	24.8	:25	15.8	:16
20	24.3	:25	28.3	1:05	19.2	:23
40	26.6	:47	32.4	1:21	23.6	:44
50	30.0	1:21	35.5	1:59	32.6	1:30
80	36.7	3:02	?	?	53.7	2:54
100	39.7	3:27.4	56.6	5:41.0	60.9	3:38

Thus, for random and for gaussian paging behavior the times are nearly identical throughout the measured range. For small programs UNIX is faster, since programs tend to get larger memory allocations without paging from the free list (as they are forced to in *VMS* when they pass their working set quota).

Other comments involving paging performance are:

a) Contrary to previous information, *VMUNIX* **does not** keep all user page tables resident in main memory at all times. The present system keeps only the pages of currently resident processes in memory. Also, the 20 April memo neglected to point out that *VMS statically allocates* space for both the kernel segment tables and the contiguous swap area on disk. This allocation is based on the maximum possible number of

* Note that *VMUNIX* **does not** move pages from disk to main memory one at a time as previously reported, nor has it done so for over a year. UNIX "pages-out" up to 16 K contiguous bytes (1 complete disk track) at a time and "pages-in" 4 Kbytes (8

processes. Thus, if we allow for a maximum of only 10 *VMS* processes (a conservative figure), of which, say, only 1 is of the quoted hypothetical 128 Mb in size, *VMS* will allocate a fixed 20 Kbytes (128Mb * 10 procs / 64 Kb) of memory for kernel segment tables, and a fixed 1.2 gigabytes (128Mb * 10 procs) of disk space, 90% of which will be unused! *VMUNIX*, on the other hand, dynamically allocates both main memory used for page tables and disk paging space. Running out of resources in *VMUNIX* may delay the completion of a process, but less total resources are required and what are required are used more effectively.

b) The "severe problem" mentioned with respect to *VMUNIX* swapping out a process when it was the only program in memory was a bug in the paging algorithm that has now been fixed.

c) The "lock-out" problem mentioned with respect to *VMUNIX* when heavy paging traffic was present was due to this same bug; again this has now been fixed.

2. **Fortran Differences.** There are several points of interest here. First to note is that the work of Rafenetti at ANL is not crucial since they have adopted UNIX there. Secondly, the cited inaccuracy of mathematical functions is not particularly decisive either. The inaccuracies of *VMUNIX*'s *DEXP()* are no worse than *VMS*'s *DLOG()* and *DLOG10()* inaccuracies. Also note that the extreme inaccuracy of X**Y in the *VMUNIX* "libnm" power function is due to a bug in the VAX hardware! (A microcode error in the "polyd" instruction, reference DEC FCO#5 for the fix to this.) *VMS* run on a system that does not have the hardware engineering change would potentially show exactly the same type of error.

We are thus left with evaluation of the "FFT" benchmarks. Here it is imperative to isolate the fortran compiler aspect of *VMUNIX* from the demand paging aspects of the system. This was obviously not done in the earlier benchmarks, since if a compute bound process fits entirely in physical memory on an otherwise idle system, then the real execution time must equal the program cpu time. Notice that this was true under *VMS* but not true under *VMUNIX*. There can *only* be two reasons for this. Either the system was not otherwise idle or the program was larger than could fit in physical memory. It is hard to conceive of a way that a 128 x 128 complex array could not fit into an idle 2 Mb machine. The FFT benchmarks were re-run on 24 April with very different results:

Table IV
Unloaded System FFT

	VMS		*"Old" UNIX Times*		*24 Apr 1981 UNIX Times*		*"New" Ratio*
	CPU	*Real*	*CPU*	*Real*	*CPU*	*Real*	
128x128	:08	:08	:26	1:43	:17	:18	1:2.25
256x256	:38	:38	2:00	7:13	1:12	1:13	1:1.92
512x512	3:43	3:44	10:32	24:59	6:00	6:09	1:1.65

These new results appear consistent with the ratio of execution speeds quoted in the past "...a factor of 1.5 to 2.5 difference between *VMS* Fortran and F77".

[Editor's Note: The UNIX times above have now been improved considerably. See "F77 Performance" by David Mosher and Robert Corbett, published in the June, 1982, issue of *;login:* (Vol. 7, No. 3).]

On a loaded system the new benchmark results can be even more impressive, although direct comparison with times quoted in the 20 April memo are not possible since no quantitative figures were given to indicate what "a loaded system" was. One reason to expect increased performance on a loaded system, however, is the fact that the FFT program exhibits strongly sequential memory accesses

K for sequential program behavior).

and thus benefits greatly from the *vadvise(SEQL)* system call.

Other comments in the area of Fortran performance are:

a) A 2.12 Mb program which exhibits sequential virtual memory accesses is far from "humble".

b) If one asks for the lowest possible priority on a machine which is not idle or otherwise running only programs of the same low priority, one should not expect to get much cpu time.

c) The "needless paging" problem mentioned with respect to "forced swapping" in *VMUNIX* was due to a previously mentioned bug in the UNIX paging algorithm. Again, this has now been fixed.

Conclusions

The picture of *VMUNIX* is not nearly as dismal as was previously reported. The various paging benchmarks quoted now run as fast as, and in the case of sequential virtual memory access even faster than, *VMS*. A kernel bug causing extraneous swapping and general system degradation during a heavy paging load has been fixed. The burden of responsibility for a large inaccuracy in a Fortran intrinsic function has been shifted from *VMUNIX* to the VAX hardware where it belongs. Future improvement in the accuracy of all math library functions on *VMUNIX* is forthcoming. The important 512 × 512 FFT benchmark performs significantly better than previously stated and, with the help of the *vadvise(SEQL)* system call and future improvements in the F77 compiler will exhibit even better performance in the not too distant future.

USENIX Association Office Report

On October 23, 1982, the new Office opened for business in El Cerrito, California (near Berkeley). Since the office has opened we have been working on the mailing list for the UNICOM registration packet and on new and renewed USENIX memberships. The UNICOM registration packets were mailed in late November. The USENIX membership data base is now up to date, with the exception of a few for which we have incomplete paperwork. All new and renewed members have been mailed the previous issues of *;login:* for this year.

Currently we are going through the paperwork for the institutional members that have not yet been sent their software distribution tape for this year. These tapes will sent out as soon as possible.

The USENIX Office can now be reached at:

USENIX Association
P.O. Box 7
El Cerrito, CA 94530

The new phone number of the Office is:

(415) 528-UNIX

The Office is staffed between 10am and 2pm, Pacific Time, weekdays. Messages may be left on a phone answering machine when no one is available.

Tom Strong
Executive Director

USENIX Association

Application for Individual or Public Membership For Calendar Year 1983

Please type or print very clearly

[ ] New

[ ] Renewal [ ] With changes to existing USENIX records

Old address: _____

_____

_____

_____

[ ] Individual Membership: $30

[ ] Non-Disclosure covered by institution with source license

[ ] Non-Disclosure covered by institution with binary license

Institutional affiliation: _____

Nature of affiliation: _____

[ ] Public Membership (Not covered by Non-Disclosure): $30

Mailing address:

Your name: _____

_____

_____

_____

_____

Phone: (_____) _____

Network address: _____

[ ] Overseas airmail, add $9.00

[ ] Invoice required, add $6.00 bookkeeping charge

Check enclosed: $_____

Return completed form to:

USENIX Association
P.O. Box 7
El Cerrito, CA 94530

You will receive a card acknowledging your membership as soon as it is processed.

Office Use Only

Inst: .....

Del: .....

Check #: .....

Mem #: .....

Date: .....

USENIX Association
P.O. Box 7
El Cerrito, CA 94530

First Class Mail

Your 1983 Individual Membership Renewal Form is Enclosed

USENIX Dues Increase

The Board of Directors, at its last meeting, voted to increase the annual dues for Individual and Public members to $30. This increase was adopted to bring membership dues more nearly in line with the cost of producing the newsletter. For members attending the USENIX conferences, the new membership discount for registration will partly offset this increase.